I claim:

1.   A method to execute a program comprising the steps of:

providing a set of arithmetic/logic (AL) instructions;

storing said set of AL instructions in at least one AL memory;

providing a set of instruction fetch (IF) instructions for programmably selecting AL

instructions to be fetched from said at least one AL memory;

providing an IF memory for storage of said set of IF instructions;

generating IF instructions and their sequencing for programmably selecting AL

instructions to be fetched from said at least one AL memory;

storing the sequence of IF instructions in said IF memory; and

fetching and executing said sequence of IF instructions to generate addresses for fetching

AL instructions from said at least one AL memory and executing the fetched AL instructions,

whereby the function of said program is accomplished.

2.   The method of claim 1 wherein the set of AL instructions are comprised of a

selected set of adds, subtracts, multiplies, divides, logical functions, shifts, rotates, permutations,

bit operations, and other arithmetic and logic type functions.

3.   The method of claim 1 wherein any instruction of said set of AL instructions

stored in said at least one AL memory can be accessed a plurality of times without duplication of

the instruction.

4.   The method of claim 1 wherein said program is comprised of a sequence of a

selected set of sequential and parallel multiple-issue function instructions embedded with a set of

control structure instructions.

5. The method of claim 4 wherein said sequential function instructions can be translated to a set of AL instructions which can be issued singly.

6. The method of claim 4 wherein said parallel multiple-issue function instructions can be translated to a set of AL instructions which can be issued in parallel.

7. The method of claim 4 wherein said set of control structure instructions are comprised of a selected set of for-do, if-then-else, case, while-do, do-until, do-while, branches, calls, returns, and auto-loop instructions.

8. The method of claim 4 wherein said program is analyzed to identify said selected set of sequential and parallel multiple-issue function instructions, said set of control structure instructions, and the sequence of instructions.

9. The method of claim 8 wherein the identified sequential function instructions, control structure instructions and the sequence of instructions are used to generate IF instructions and their sequencing for programmably selecting AL instructions to be fetched from said at least one AL memory for execution.

10. The method of claim 8 wherein said set of AL instructions are stored in at least two AL memories, and wherein the identified parallel multiple-issue AL instructions, control structure instructions, and the sequence of instructions are used to generate IF instructions and their sequencing for programmably selecting AL instructions to be fetched from said at least two AL memories for execution when parallel multiple-issue AL instructions are indicated.

11. The method of claim 8 whereby said program is further analyzed to reduce the size of said at least one AL memory, comprising the steps of:

identifying single and duplicate AL instructions in a section of code making up said program;

46

removing all but one of the duplicate AL instructions from said at least one AL memory;

identifying at what addresses in the program sequence the duplicate AL instructions occur and the address of the single reference AL instruction; and

using this identification in generating IF instructions and their sequencing such that whenever a duplicate AL instruction is required, an IF instruction is executed to create the address for the single reference AL instruction stored in said at least one AL memory, whereby a single AL instruction is stored in said at least one AL memory instead of a plurality of duplicate AL instructions for said section of code making up said program.

12. A processor system comprising:

an instruction fetch (IF) memory storing a sequence of IF instructions;

a programmable instruction fetch mechanism comprising means to fetch and execute IF instructions;

at least one arithmetic/logic (AL) instruction memory (IMemory) storing a set of AL instructions, and

at least one AL decode and execute unit, said programmable instruction fetch mechanism operating to fetch IF instructions from said IF memory and executing the fetched IF instructions thereby generating IMemory instruction addresses to select AL instructions to be fetched from at least one IMemory for execution on the at least one AL decode and execute unit.

13. The processor system of claim 12 wherein a first IF instruction type comprises an opcode field specifying a sequential fetch operation and an IMemory address field, and a second IF instruction type comprises the fields of said first IF instruction plus an additional field indicating the number of instructions to be sequentially fetched and specifying the IMemory address as the starting address for a group of instructions.

47

14.   The processor system of claim 13 wherein the first and second IF instruction types each comprise an additional field for an IF memory instruction address.

15.   The processor system of claim 13 wherein the first and second IF instruction types each comprise at least one additional field for a conditional branch address specifying a location in the IF memory.

16.   The processor system of claim 13 wherein the first and second IF instruction types each comprise two additional fields for a loop count and a loop end address; the address of the IF instruction identifying the loop start address which together with said loop end address identifies the program loop and the loop count controls the number of iterations of the loop.

17.   The processor system of claim 12 further comprising:

a third IF instruction type for parallel multiple-issue instructions;

an additional arithmetic/logic (AL) instruction memory (IMemory) comprising a set of AL instructions; and

an additional AL decode and execute unit, said programmable instruction fetch mechanism operating to fetch IF instructions from said IF memory and when executing a third IF instruction type generating at least two IMemory instruction addresses to select AL instructions to be fetched from at least two IMemories for execution on the at least two AL decode and execute units.

18.   The processor system of claim 17 wherein said third IF instruction type comprises at least three fields: an opcode field specifying a parallel multiple-issue fetch operation, and at least two IMemory addresses.

19.   The processor system of claim 18 wherein the third IF instruction type comprises an additional field for an IF memory instruction address.

20. The processor system of claim 17 wherein the third IF instruction type comprises at least one additional field for a conditional branch address specifying a location in the IF memory.

21. The processor system of claim 17 wherein the third IF instruction type comprises two additional fields for a loop count and a loop end address; the address of the IF instruction identifies the loop start address which together with said loop end address identifies the program loop and the loop count controls the number of iterations of the loop.

22. The processor system of claim 17 wherein a fourth IF instruction type comprises at least five fields:

a load IMEM instruction opcode;

at least two base address register indicator bits;

at least one IMemory offset; and

at least one data memory offset.

23. The processor system of claim 12 further comprising:

a fifth IF instruction type for parallel multiple-issue instructions;

an additional arithmetic/logic (AL) instruction memory (IMemory) comprising a set of AL instructions;

an additional AL decode and execute unit; and

an IMemory and its associated decode and execute units separate from the other IMemories associated with the AL decode and execute units, said programmable instruction fetch mechanism operating to fetch IF instructions from said IF memory and when executing a fifth IF instruction type generates at least three IMemory instruction addresses to select

instructions to be fetched from at least three IMemories for execution on the at least two AL decode and execute units and on the separate decode and execute unit.

24. The processor system of claim 23 wherein the fifth instruction type comprises at least three fields:

an opcode field indicating a multiple instruction fetch operation;

at least one IMemory address field specifying a common address for the at least two IMemories; and

a separate IMemory address field specifying an address for a separate IMemory.

25. A processor system comprising:

an instruction fetch (IF) memory comprising a sequence of IF instructions;

a programmable instruction fetch mechanism comprising means to fetch and execute IF instructions;

at least two IMemory address bus interfaces between the programmable instruction fetch mechanism and at least one processing element (PE); and

at least one PE further comprising:

at least two arithmetic/logic (AL) instruction memories (IMemories) which interface with the at least two IMemory address buses;

at least two AL decode and execute units; and

a set of address registers to be used for addressing operations, wherein said programmable instruction fetch mechanism operating to fetch IF instructions from said IF memory and executes the fetched IF instructions thereby generating IMemory instruction addresses to select processor element (PE) AL instructions singly from one of the AL instruction memories for execution on one of the AL decode and execute units.

26. The processor system of claim 25 further comprising operation steps to select at least two PE AL instructions for execution on the at least two AL decode and execute units.

27. The processor system of claim 25 further comprising at least two processor elements controllable as one concatenated processor element with a first type IMemory AL instruction specifying a concatenated operation, and controllable as two independent processor elements with a second type IMemory AL instruction specifying at least two independent operations.

28. The processor system of claim 27 wherein the second type IMemory AL instruction is a subset of the first type IMemory AL instruction.

29. The processor system of claim 25 wherein the PE AL instructions comprise multiple PE AL instruction formats including an optional vector parameter field and an optional conditional execution field.

30. The processor system of claim 25 wherein a PE AL instruction format comprises an opcode field, a data type field, a data memory selection field, and at least one operand field.

31. The processor system of claim 25 wherein a PE AL instruction format comprises an opcode field, a vector parameter field, a vector address register field, a data memory selection field, and at least one operand field.

32. The processor system of claim 25 wherein a PE AL instruction format comprises an opcode field, a vector parameter field, a data type field, a data memory selection field, and at least one vector operand parameter field, at least one address register field, and at least one operand offset field.

33. The processor system of claim 25 wherein a vector operation executes on at least one PE, the system further comprising an execution sequence of initiating a vector operation

51

when a PE AL instruction format supporting vector setup operation executes, causing at least one operand address to be loaded into an address register and starting the vector operation each time a PE AL instruction format supporting vector system is fetched and executed.

34. The processor system of claim 25 wherein a PE AL instruction format comprises an opcode field, a data type field, a data memory selection field, at least one operand field, and a 16-bit immediate field.

35. The processor system of claim 25 wherein a PE AL instruction format comprises an opcode field, a data type field, a data memory selection field, at least one operand field, and a 32-bit immediate field.

36. The processor system of claim 25 wherein the PE AL instructions of a first IMemory AL instruction type are comprised of multiple 64-bits formats.

37. The processor system of claim 36 wherein the PE AL instructions of a second IMemory AL instruction type are comprised of multiple 32-bit formats.

38. The processor system of claim 37 wherein the PE AL instructions of a third IMemory AL instruction type are comprised of multiple 16-bit formats.

39. The processor system of claim 38 wherein the 16-bit format PE instructions comprises a target register specified as a function of one of the source operand fields.

40. The processor system of claim 25 further comprising at least two clusters of two processor elements each controllable as two processor elements with two different first type IMemory AL instructions, and controllable as four independent processor elements with four different second type IMemory AL instructions.

41. The processor system of claim 25 further comprising at least two clusters of four processor elements each controllable as two processor elements with two different first type

52

IMemory AL instructions, controllable as four independent processor elements with four different second type IMemory AL instructions, and controllable as eight independent processor elements with eight different second type IMemory AL instructions.

42.   A processor system comprising:

an instruction fetch (IF) memory comprising a sequence of IF instructions;

programmable instruction fetch mechanism comprising means to fetch and execute IF instructions;

at least one arithmetic/logic (AL) instruction memory (IMemory) comprising a set of AL instructions; and

at least one AL decode and execute unit, said programmable instruction fetch mechanism operating to fetch IF instructions from said IF memory and executing the fetched IF instructions thereby generating IMemory instruction addresses to select AL instructions to be fetched from at least one IMemory for execution on the at least one AL decode and execute unit and generating an address for the next IF instruction.